LIGHT ATTENUATION AND SHADOWS

Template 3

Computer Graphics 2

What’s New?

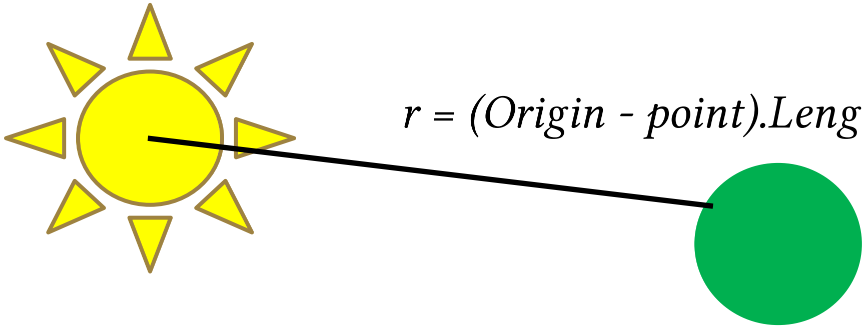
- we consider attenuation of the light
- we want to render shadows

➔ change Rendering -- Camera

```
public Boolean UseShadows = true;
public Boolean UseLightAttenuation =  true ;
```

Light Attenuation

➡ change Lightening -- PointLight



$r = (Origin - point).Length;$

□ function of light attenuation depends on the distance from the light source

$$f(r) = \frac{1}{1 + a^{*}r + b^{*}r^{2}}$$

□ we use $a = 0.02\ ,\ b = 0.0$

□ if UseLightAttenuation = false then

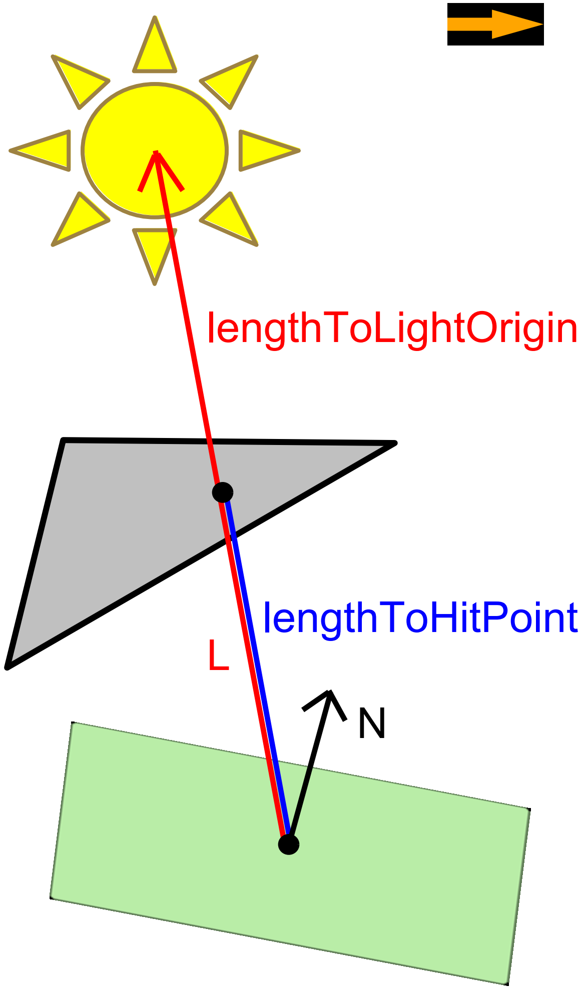
Phong Shader

- change Shading -- Phong

$$I = k_a I_a + \sum_{i=1}^{n} \left(f(r)\, k_d I_{i,d} (\boldsymbol{l_i} \cdot \boldsymbol{n}) + f(r)\, k_s I_{i,s} (\boldsymbol{r_i} \cdot \boldsymbol{v})^{n_s} \right)$$

```
public override Vector3 GetColor( . . . , Double attenuation , . . .)
```

Shadows



➡ change Rendering -- Camera

- □ if UseShadows = true and $N.L > 0$
  we must check if the point is in shadow

  - □ default value

    inShadow = false

  - □ if (lengthToHitPoint < lengthToLightOrigin)

    inShadow = true

Final Color

➡ change Rendering -- Camera

- in the Template 2, we had

```
hitPointColor += ray.HitModel.Shader.GetColor(...);
```

- now, we have
  - if the point is not in shadow

```
hitPointColor += 1.0* ray.HitModel.Shader.GetColor(...);
```

  - if the point is in shadow

```
ambientIntensity = ray.HitModel.Shader.GetAmbientColor(hitPoint.Position).Length;
hitPointColor += ambientIntensity* ray.HitModel.Shader.GetColor(...);
```

Questions?